Student Name: **Stud id:** **sect #:** **serial#:**

University of Bahrain College of Information Technology
Department of Computer Science

ITCS332: Organization of Programming Languages **Quiz #6: Assignment + Subprograms**

****

1) Name two kinds of operators used in expressions: *Arithmetic* and *Logical operators*.
2) In expressions, the side effects may occur when a function changes *non-local variables* or *two-way parameters.*
3) For every subprogram call, the system creates *Activation Record* on the run-time stack. The expression that allows using operators of different type is called *mixed mode* expressions.
4) The implementation of computations specified by an expression causes two actions: *Fetching the operands from memory* and *Executing the operations on those operands*.
5) For every subprogram call, the *RETURN ADDRESS* and *PARAMETERS* are stored on the activation record.
6) The format and the structure of the activation record are specified by *Compiler Constructor*. In subprograms, the parameter passing methods are selected by *Language Designer* .

Carefully study the following C-like code and answer the next 2 questions

```
int fun (int *k)
{
  *k += 5 ;
  return 2 * (*k) - 5 ;
}
void main()
{
    int i = 6, j = 4, sum1, sum2 ;
    sum1 = ( i / 2 ) + fun ( &i);
    sum2 = fun (&j ) + (j / 2);
}
```

What are the values of `sum1` and `sum2` if the operands in the expressions are evaluated **right to left**?

```
sum1 = i/2+fun(&i)
     = i/2+fun(6)
     = i/2+ 2*11-5
     = 11/2 + 17
     = 22
```

```
sum2  = fun(&j)+ j/2
      = fun(4)+ 4/2
      = 2*9-5 + 2
      = 13 + 2
      =  15
```

Student Name: **Stud id:** **sect #:** **serial#:**

University of Bahrain College of Information Technology
Department of Computer Science

ITCS332: Organization of Programming Languages **Quiz #6: Assignment + Subprograms**

****

7) The expression evaluation process depends on *the order of operators* evaluation and the *order of operands* evaluation.

8) The two types of notations used to write the binary operators are: *infix* and *prefix* notations.

9) Name two kinds of errors that may occur in evaluating expressions are *Type errors* and *Overflow.*

10) A Procedure abstracts a program unit into *a standalone statement*; while a function abstracts a program unit into *an expression (part of a statement).*

11) The binding of actual parameters to formal parameters is done according to their *position* or *names.*

12) The starting address of the activation record is defined by *static link* and the ending address is defined by *dynamic link.*

Carefully study the following C-like code and answer the next 2 questions

```
int fun (int *k)
{
  *k += 5 ;
  return 2 * (*k) - 5 ;
}
void main()
{
    int i = 6, j = 4, sum1, sum2 ;
    sum1 = ( i / 2 ) + fun ( &i);
    sum2 = fun (&j ) + (j / 2);
}
```

What are the values of `sum1`and `sum2` if the operands in the expressions are evaluated **left to right**?

```
sum1 = i/2+fun(&i)
     = 6/2+fun(6)
     =  3 + 2*11-5
     =  3 + 17
     = 20
```

```
sum2  = fun(&j)+ j/2
      = fun(4)+ j/2
      = 2*9-5 + 9/2
      = 13 + 4
      =  17
```